522027 IN-04
117766
P-24

NASA Technical Memorandum 102882

A Workstation-Based Evaluation of a Far-Field Route Planner for Helicopters

David N. Warner, Jr. and Francis J. Moran

(NASA-TM-102882) A WORKSTATION-BASED EVALUATION OF A FAR-FIELD ROUTE PLANNER FOR HELICOPTERS (NASA) 24 p

N92-33609

Unclas

G3/04 0117766

June 1991

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 102882

A Workstation-Based Evaluation of a Far-Field Route Planner for Helicopters

David N. Warner, Jr. and Francis J. Moran, Ames Research Center, Moffett Field, California

June 1991

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

SUMMARY

Helicopter flight missions at very low, Nap of the Earth, altitudes place a heavy workload on the pilot. To aid in reducing this workload, Ames Research Center has been investigating various types of automated route planners. As part of an automated preflight mission planner, a route planner algorithm aids in selecting the overall (far-field) route to be flown. During the mission, the route planner can be used to replan a new route in case of unexpected threats or change in mission requirements. This report describes an evaluation of a candidate route planning algorithm, based on dynamic programming techniques. This algorithm meets most of the requirements for route planning, both preflight and during the mission. In general, the requirements are to minimize the distance and/or fuel and the deviation from a flight time schedule, and must be flyable within the constraints of available fuel and time.

INTRODUCTION

Helicopter flights are frequently flown in areas and under hazardous conditions where a heavy workload is placed on the pilot. One example is a military mission flown at very low altitudes, from about 50 m above treetop level to about 2 - 6 m above ground level. The latter elevation is known as Nap of the Earth, or NOE. Successful operations in this flight regime requires automation to aid the pilot, both in preflight mission planning and in the cockpit. This area of automation has been a major effort at NASA, Ames Research Center for several years (refs. 1-6). Part of this effort is the study of automated route planners, which fit into three categories: far-field, mid-field, and near-field. Far-field planners determine the overall route to be flown, e.g., which valley to fly through, by having knowledge of the terrain based on a coarse, digital terrain database (refs. 3,4). Mid-field route planners, utilizing the widely spaced waypoints and closely spaced digital terrain data, plan a safe route through the terrain (refs. 3,5). Obstacles which do not appear in the digital database are dealt with by a near-field planner or guidance system, which relies on on-board sensors for its inputs (ref. 6).

A mission planner is used by the mission commander for pre-flight mission planning. It is typically an interactive computer tool which allows the user to view a graphical map of the operational area and to input locations of the destination location, any intermediate waypoints or destinations, known man-made threats, weather disturbances, weapons stores, fuel capacity, time constraints or goals, and any other mission goals. Part of the operation of this mission planner is computation of the "best" route by a far-field route planner. A route planner of this type could also be used by a pilot during a flight mission to compute a new route if the preplanned one becomes inappropriate as new information is received. This report describes a qualitative evaluation of a candidate route planner algorithm.

The far-field route planner described in this report was provided to Ames Research Center as part of a Small Business Innovative Research (SBIR) contract. As designed and implemented by the contractor on a personal computer (ref. 5), a graphical user interface gave the system many features of a full mission planner. Although parts of the graphical user interface were implemented at Ames

on a SUN workstation along with the route planner software, the primary emphasis at Ames has been to use and evaluate the route planning algorithm independent of other higher-level mission planning aspects. This report describes this route planner in general terms, because of proprietary restrictions, and its performance as tested on a SUN workstation. Also included is a brief description of the graphical user interface which was used on the workstation to define inputs to the planner, execute the route planner algorithm, and display the resulting route.

FEATURES DESIRED IN A ROUTE PLANNER

A route planner has one overall function to perform, that is to compute an acceptable route from a designated starting location to one or more optional, intermediate goal locations and to a designated final destination. The criteria which defines the acceptability of the route depends on the context of the route planning and some criteria which must be met. For military helicopter NOE missions, a planned route must ensure that a helicopter will arrive at its destination with a high probability of survival. In general, an acceptable route minimizes the distance and/or fuel and deviation from a flight time schedule, and must be flyable within the constraints of available fuel and time. Some missions include the requirement to meet time goals at specified locations, for example to rendezvous with other aircraft or to pass a control location. Also, the algorithm should run fast enough (within about one minute) on an airborne computer to be used for route replanning during the mission.

In order to accomplish these requirements, a far-field planner must have knowledge of the terrain and known threats, either weather to be avoided or man-made threats such as missile sites. Since a far-field planner only determines a general route to be flown, the terrain can be defined by a relatively coarse grid of data, on the order of 0.25 to 2 km between data points depending on the terrain's topology. Threats can be made known to the planner by specifying their locations and a model which defines their range and other pertinent parameters. From the model and the terrain topology at the threat location, terrain masking of the threat can be used to maximize the probability of survival. Some capability must be made for user input of starting and ending locations as well as any intermediate locations such as rendezvous locations or targets which are along the route. Once the inputs are known, a computerized algorithm can be used to plan a route which meets the criteria for acceptability.

GRAPHICAL USER INTERFACE

Inputs to and control of the route planner algorithm and display of the resulting planned route are provided by the graphical user interface, figure 1. This user interface is a special purpose window on a SUN workstation. Along the top of the upper subwindow are buttons used to select various functions. From left to right, the top row of buttons start the route planner algorithm, plot results in variable vs distance form, select various map options, control display of a grid on the map, clear user-added features from the map, select prestored scenarios, control a route weighting function, and quit the program. The icons on the second row are used to select types of control points or threats for placement on the map by the user. The icons represent, from left to right, the route starting location,

a non-route-optimized route control point, an optimized route control point, a target, the destination, an adverse weather region, a missile site, an antiaircraft gun location, and a radar site. Route control points as used in this report are locations which the user desires that the route pass over. These locations may be rendezvous points, navigation update locations, passenger drop off sites, or may be specified to force the route through areas the planner would not ordinarily select. The selection buttons and icons are used by moving the arrow-shaped cursor onto the desired button or icon by moving a mouse and pressing ("clicking") the left mouse button. The user then moves the cursor to place the selected icon on the colored map in a lower subwindow (described shortly) and again clicking the left mouse button. When this is done, the icon is displayed on the map, and its location is saved for use by the route planner algorithm.

The second subwindow is where route length and number of waypoints along the computed route are displayed. To the right is a small subwindow where the route planner algorithm is identified and map identifier and scaling are displayed. Also shown are the current location of the cursor when it is moved in the map subwindow. At the bottom of the subwindow is the current value of the route weighting factor, to be described in a later section.

The operational area of interest is displayed in the large graphics subwindow. Terrain elevations on the map are color coded as shown in the smaller subwindow below the map, where blue is sea level and the other colors represent increasing altitude segments from lowest (light green) to highest (dark maroon) in the area. Route control points and threat icons are displayed at locations selected by the user. If desired the user can click on the "GRID" button to display grid lines on the map.

To plan a route, the user first selects the proper map. Then the starting location, intermediate route control points, target(s) location, and the final destination are positioned on the map. Locations of any known threats are positioned on the map and red lines are automatically drawn to show the threat region covered. Then the user clicks on the "ROUTE PLANNER" button to execute the route planner. The route planner algorithm is executed and the computed route displayed as a white line. Plots of terrain altitude, cost, and cumulative cost along the route can be displayed by clicking on the "PLOT" button. Prestored scenarios can be selected by clicking on the "FILE" button.

ROUTE PLANNER DESCRIPTION AND EVALUATION

General Description

The route planner, called Dynaplan by the developer, is based on a dynamic programming algorithm (ref. 3). The implementation by the contractor departs from more standard implementations only in a few areas, such as terrain data handling, threat definition and masking, and manipulations of the cost data to reduce computation speed. Because of the proprietary nature of the software, these features will be described below only in general terms. Reference 1 describes the route planner and also describes some mission planning aspects of the contractor's implementation which were not used in the Ames system.

The general procedures used in implementing the dynamic programming algorithm to compute a near-optimum route is as follows. First, an array which defines the flight environment is initialized to the terrain elevation. Then the threat models selected and located by the user are superimposed on the terrain array. The three-dimensional transition cost array is generated for eight control directions at each cell by computing the cell-to-cell transition times. These transition times are approximations since cell-to-cell distance is an approximation and the ground speed is assumed constant. After the state space parameters defining the operational situation are thus defined, the dynamic programming algorithm is executed to compute the near-optimum route. A loop structure is used to compute the route as separate near-optimum segments from the starting location through the intermediate control-points and targets, if any, to the final destination. Some of these computational steps will be described more below as results of the evaluation is described and illustrated.

Terrain Data

To represent the terrain in the dynamic programming grid, digital terrain elevation data (DTED) used for this study was obtained from the U. S. Geological Survey. It is a one degree by one degree area in northern California, called here the Napa data. The spacing between data points is three arc-seconds, which is approximately 100 m north-south and 80 m east-west. For testing of the route planner algorithm, an area 80 × 80 km was selected, so a subset of the original 1201 × 1201 elevation data points were used. The elevation data points for this area were reduced by interpolation to 480 × 480 data points for display, as shown in figure 1. Since terrain data at this resolution (167 m) is not deemed necessary for far-field route planning, the data was further compressed for test purposes to 40 × 40 (2 km spacing), 80 × 80 (1 km), 160 × 160 (0.5 km), and 240 × 240 (0.33 km). The altitude was set to 75% of the maximum to minimum elevations in each subarea, or cell, figures 2-5 based on empirical evaluations.

The dynamic programming cells are initialized to the compressed terrain elevations. These values may be elevation in feet or meters or in some form of scaled units to minimize computer memory and disk storage requirements. The contractor normalized the elevations to 0 to 255 to correspond to one byte to minimize computer memory requirements and to reduce program execution time. Normalized terrain data was used for these tests as well as elevations in meters to evaluate any effects on the algorithm.

Threats

Threats implemented in the test system include adverse weather, a missile, an antiaircraft gun, and a radar site. Generic models for each threat were used since selection of different missile types, for instance, was not required to evaluate the route planner algorithm. The threat models are simply values whose magnitude is a relative value of the threat lethality and which decreases linearly to the defined ranges. Ranges used are listed below in table 1. Threat masking by the terrain is accomplished when the threat model is applied to the appropriate dynamic programming cost cells which were initialized to the terrain elevations. A separate threat intervisibility algorithm was used to display the lethal areas around threats. Examples of the threat displays and their effects are shown in figure 6.

Table 1. Threat ranges

Threat	Range (km)
Anti-aircraft gun	2
Missile	6
Thunderstorm	8

Transition Costs

The cell-to-cell transition cost computed by the planner algorithm is a function of elevation change and distance from the current cell to the adjacent cell. Various ways of implementing the cost function are possible. As delivered, the transition cost was computed as the product of the average elevation change and the transition distance,

$$C_t = E_t * D_t$$

where C_t is the cell-to-cell transition cost, E_t is the average elevation of the current and adjacent target cells, and D_t is the distance from center to center of the two cells. To reduce computer memory requirements as previously mentioned, terrain elevations were normalized to the range of 0 to 255. Another simplification was to use 100 as north-south and east-west cell transition distances and 141 as the diagonal distances. For this test, the terrain data had actual cell transition distances which varied from 0.33 to 2 km and actual terrain elevations from 0 to 1427 m. This gave an equivalent elevation quantization of 5.6 meters. This cost function penalizes elevation changes more than distance with the result that the algorithm favors lower elevation routes whenever possible; i.e., it is a valley-seeking algorithm. An example of the algorithm's operation using normalized, 1 km spaced terrain data is shown in figure 7. The route chosen followed the lowest elevation possible, a distance of about 117 km.

In order to assess the effects of the elevation and distance approximations, the data used for the parameters was changed to be correct measures of elevation and distance in meters. Using the new numbers gave the result shown in figure 8. The route appears identical to the first one shown in figure 7. Examination of numerical data from the computations confirmed that only very minor differences were present. This result confirms that some approximations in cost computations are valid. Because of a lack of adequate time, an exhaustive verification was not done however.

In many situations, a long route like that shown in figure 7 is undesirable if a shorter route is available with only small altitude changes required. The pilot can influence the algorithm's route computation by visually examining the terrain display and placing an intermediate control-point at some appropriate location. Figure 9 shows an intermediate control-point placed to force the route through a different valley, resulting in a distance of only about 36 km.

Although the pilot should have the capability to place control-points at arbitrary locations as described above, a more general method of weighting the route calculation between minimizing distance and minimizing altitude is desired. One rather simple method to do this was tested using a weighting factor W_{ED} to modify the cell-to-cell transition cost to

$$C_t = \{(10 - W_{ED}) * E_t\} + (W_{ED} * D_t) / 10$$

where the weighting factor W_{ED} is adjustable by the pilot from 0 for minimum elevation to 10 for minimum distance. Figure 10 shows three results of varying W_{ED}. The longest route (114.45 km) is for $W_{ED} = 0$. The shortest route (33.74 km), $W_{ED} = 10$, produced the undesirable effect of going over the highest terrain in the area. Values from 2 to 8 all gave short valley routes (about 34 km). Only minor differences were noted for values of 2, 4, 6, and 8, indicating that a better implementation of the equation might give more control of weighting. Nevertheless, the test proved that the elevation versus distance weighting concept can provide some user control of route characteristics.

Effects of Terrain Data Spacing

To investigate the effects of varying terrain data spacing on the route planner algorithm, a scenario representing an operational situation with a defined target was used, figure 11(a). No threats were included to allow the algorithm to compute route segments through the terrain unhindered by man-made effects. The computed route goes from the starting location through the low valleys to the target and then to the final destination. As described earlier, the route is computed in two segments. The first segment is from the starting location to the target; the second, from the target to the final destination.

For the first test case, the route was computed using terrain data in a 240 × 240 grid spaced at 0.33 km, figure 11(a). The map display is 480 × 480. The route distance is 54.5 km. Figure 11(b) shows this route drawn on the terrain displayed at the same 240 × 240 grid as the planner algorithm used. This displayed map and route are almost identical to that shown in figure 11(a). Visual inspection of the route shows it generally following the lowest terrain, as expected, from the starting location to the target in a southeasterly direction. Between the target and the final destination, the route takes a fairly direct path over a low rise and down through a pass between two higher areas. This route segment chosen was unexpected, since a lower route is available a short distance to the north. The chosen route had a lower cost due to the small terrain elevation change for a short distance than the somewhat longer, but lower altitude, route would have had. Elapsed compute time for this test was 125 seconds, including the UNIX operating system (SunOS) overhead.

Using 0.5 km (160 × 160) spacing terrain data gives the results in figure 12. Comparing figure 12(a) to figure 11(a) shows that the first route segment is offset a small distance from the lowest terrain elevation. The route distance is 54.3 km. In figure 12(b), the route still appears to be very close to the lowest terrain elevation. Elapsed compute time was 50 seconds.

When the terrain data spacing was further increased to 1.0 km (80 × 80), the route offset becomes more pronounced when viewed on a high resolution display, as shown in figure 13(a). The route distance is 54.0 km. When viewed at the same resolution as the planner algorithm used, the route

appears to be still roughly centered in the lowest part of the terrain, as shown in figure 13(b). Elapsed compute time was 10 seconds.

These route offsets are consequences of less accurate representation of the terrain. A different algorithm or weighting factor for reducing the terrain data grid size would probably change the offset somewhat, but some offset is inevitable, since a relatively large area is processed to determine a cell's weighted elevation. Operationally, this offset would likely be of minor importance, as is the small distance differences among the routes. Lower level route guidance systems using high-resolution terrain data could, depending on their algorithms, move the route closer to the lowest elevation. Pilots or other persons using the route planner may find the offset objectionable, however if they view the map as a high resolution image.

Another consequence of increasing the terrain data spacing is the reduced knowledge of the terrain. As closely spaced data is averaged to reduce the grid size, knowledge of narrow valleys suitable for use by helicopters is lost. This loss reduces the route planner's capability for finding routes to help minimize probability of detection.

SUMMARY OF RESULTS

The far-field route planner which has been described has most of the features desired in a route planner of this type. It computes generally acceptable routes to user specified control locations and targets and to a final destination. The routes avoid threats whenever possible and appear to minimize the cost criteria programmed into the algorithm, i.e., the routes appear to follow lower altitudes whenever possible but generally will allow moderate altitude changes to avoid long excursions around high regions. Occasionally however, routes are computed which appear to be excessively long when shorter routes with moderate altitude increases are available. The cursory testing of an alternate cost function indicates that a different implementation of the cost function could provide some improvement.

The planner algorithm appears to make good use of the terrain elevation data and to efficiently integrate knowledge of threats into the cost matrix. The question about what resolution of terrain elevation data is required has not necessarily been answered by this study, since it is so dependent on the terrain characteristics and the mission and route requirements. However the planner algorithm computes a route in 125 seconds or less for resolutions of 0.33 to 1 km terrain data spacing, which should be acceptable for a far-field route planner.

One area where the tested algorithm lacks some of the requirements is in assessing, or providing information to assess, a route's capability to allow a helicopter to meet time and fuel constraints. Perhaps testing whether a route meets constraints would best be performed by a higher level mission planner, but it would need information from the route planner. This route planner algorithm computes flight time and fuel usage as linear functions of route distance, which is horizontal distance only and does not account for altitude changes. This simplification may not be adequate to assess whether any specific route meets mission planner defined time or fuel constraints, but was not studied in this evaluation.

REFERENCES

1. Swenson, H. N.; Paulk, C. H., Jr.; Kilmer, R. L.; and Kilmer, F. G.: Simulation Evaluation of Display/FLIR Concepts for Low-Altitude, Terrain-Following Helicopter Operations. AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting, Colorado Springs, CO, October 14-16, 1985.

2. Clement, W. F.; McRuer, D. T.; and Magdaleno, R. E.: Some Data Processing Requirements for Precision Nap-of-the-Earth (NOE) Guidance and Control of Rotorcraft. NASA CR-177453, February 1987.

3. Denton, R. V.; Pecklesma, N. J.; and Smith, F. W.: Autonomous Flight and Remote Site Landing Guidance Research for Helicopters. NASA CR-177478, August 1987.

4. Berger, K. M.; Abramson, M. R.; and Deutsch, O. L.: Far-Field Mission Planning For Helicopters. NASA CR-177560, August 1990.

5. Pekelsma, N. J.: Optimal Guidance with Obstacle Avoidance for Nap-of-the-Earth Flight. NASA CR-177515, December 1988.

6. Cheng, V. H. L.: Concept Development of Automatic Guidance for Rotorcraft Obstacle Avoidance. IEEE Transactions on Robotics and Automation, vol. 6, no. 2, April 1990.

7. Larson, R. E.; and Casti, J. L.: Principles of Dynamic Programming. Marcel Dekker, Inc., New York.

ORIGINAL PAGE
COLOR PHOTOGRAPH

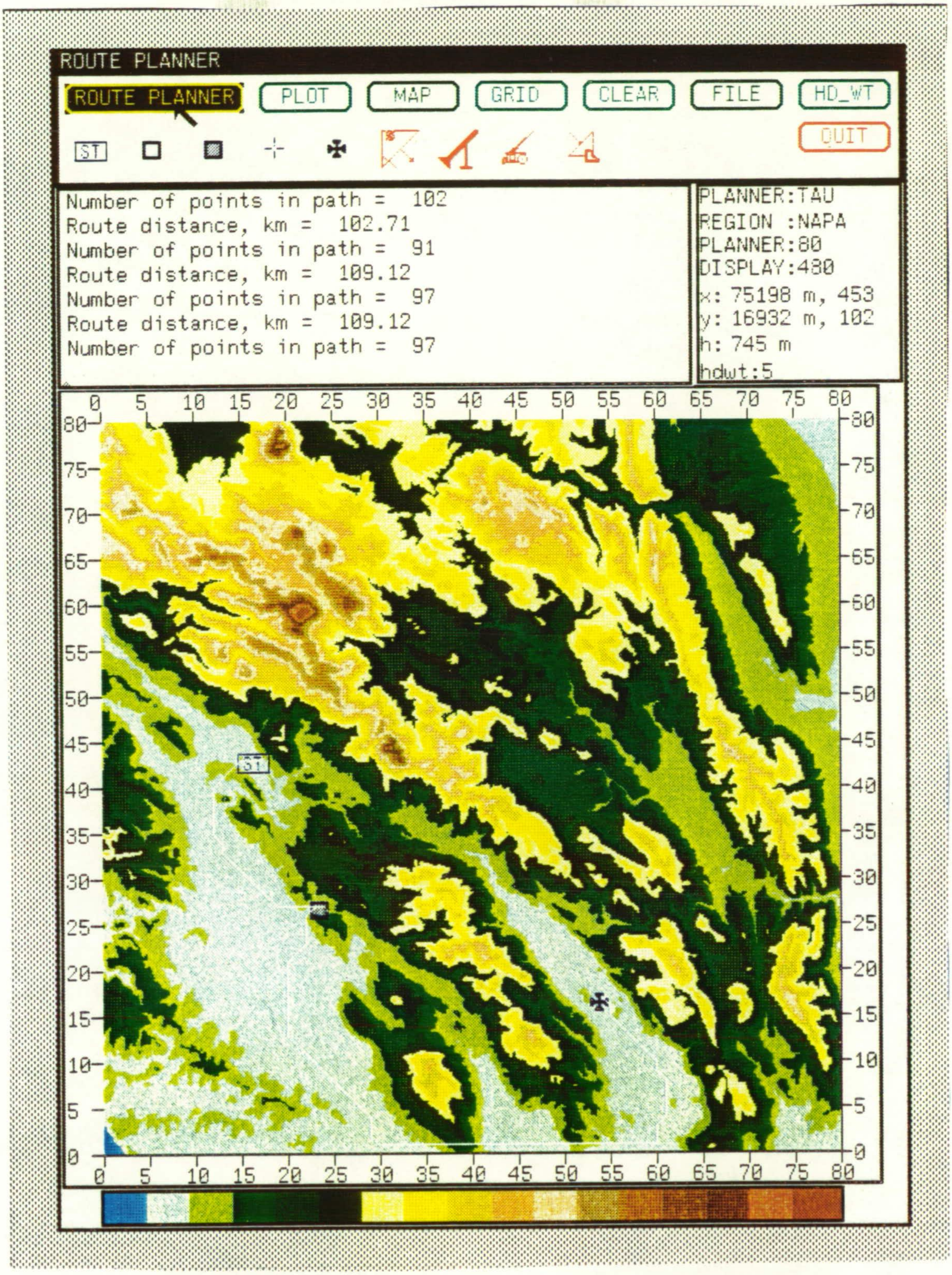

Figure 1. Graphical user interface for route planner.

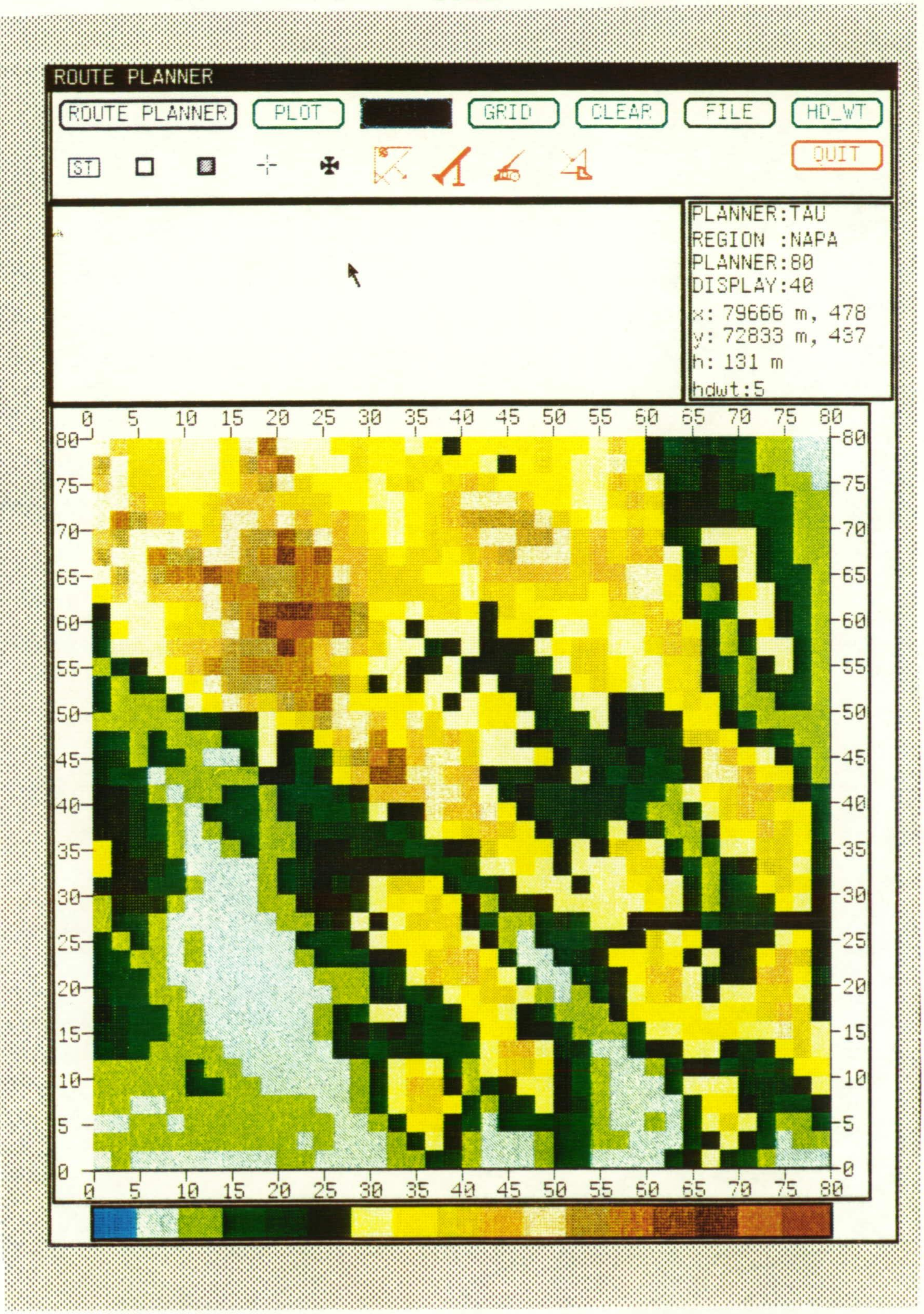

Figure 2. Graphical representation of 40 × 40 (2.0 km) grid terrain as used by route planner.

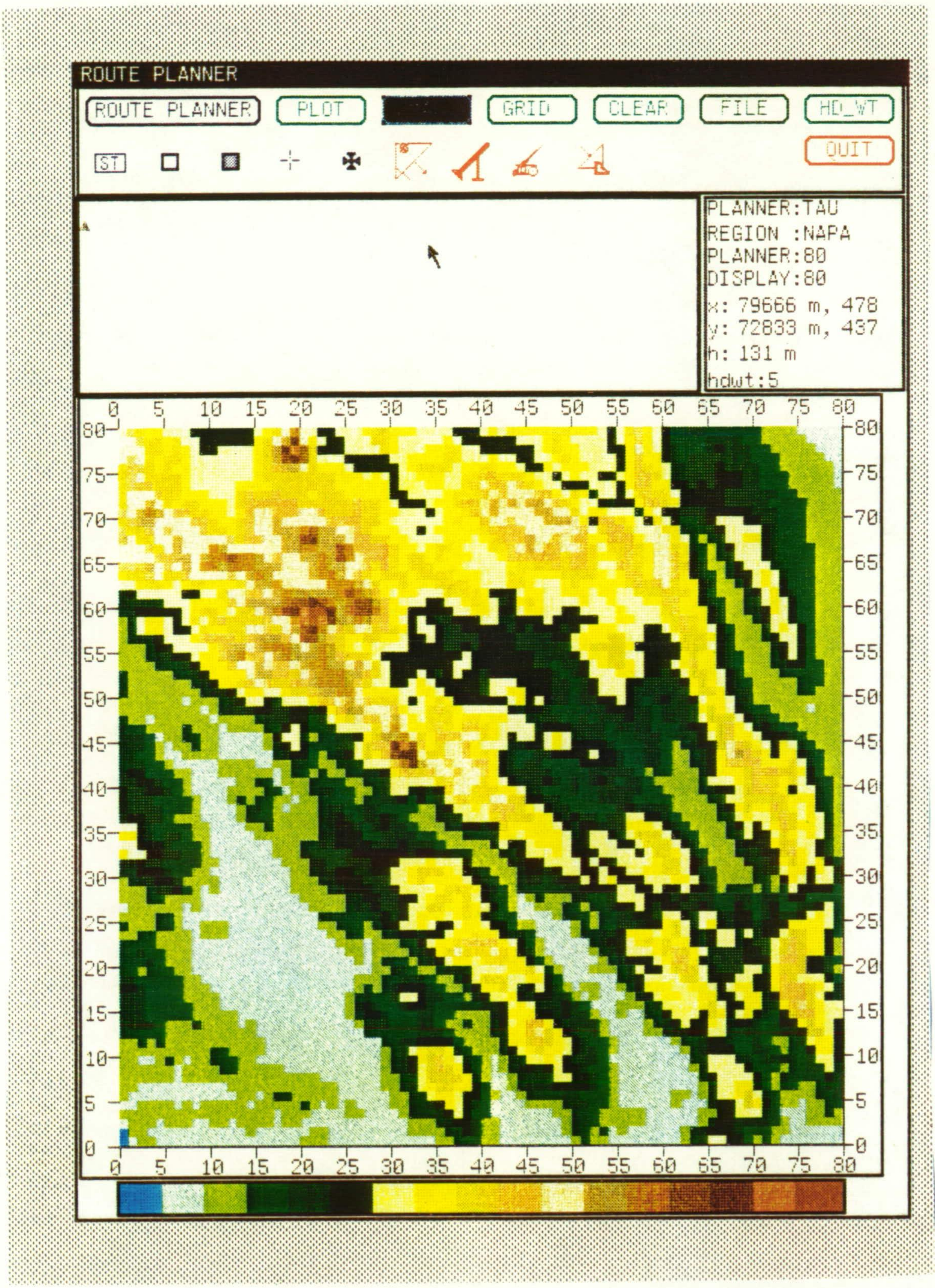

Figure 3. Graphical representation of 80 × 80 (1.0 km) grid terrain as used by route planner.

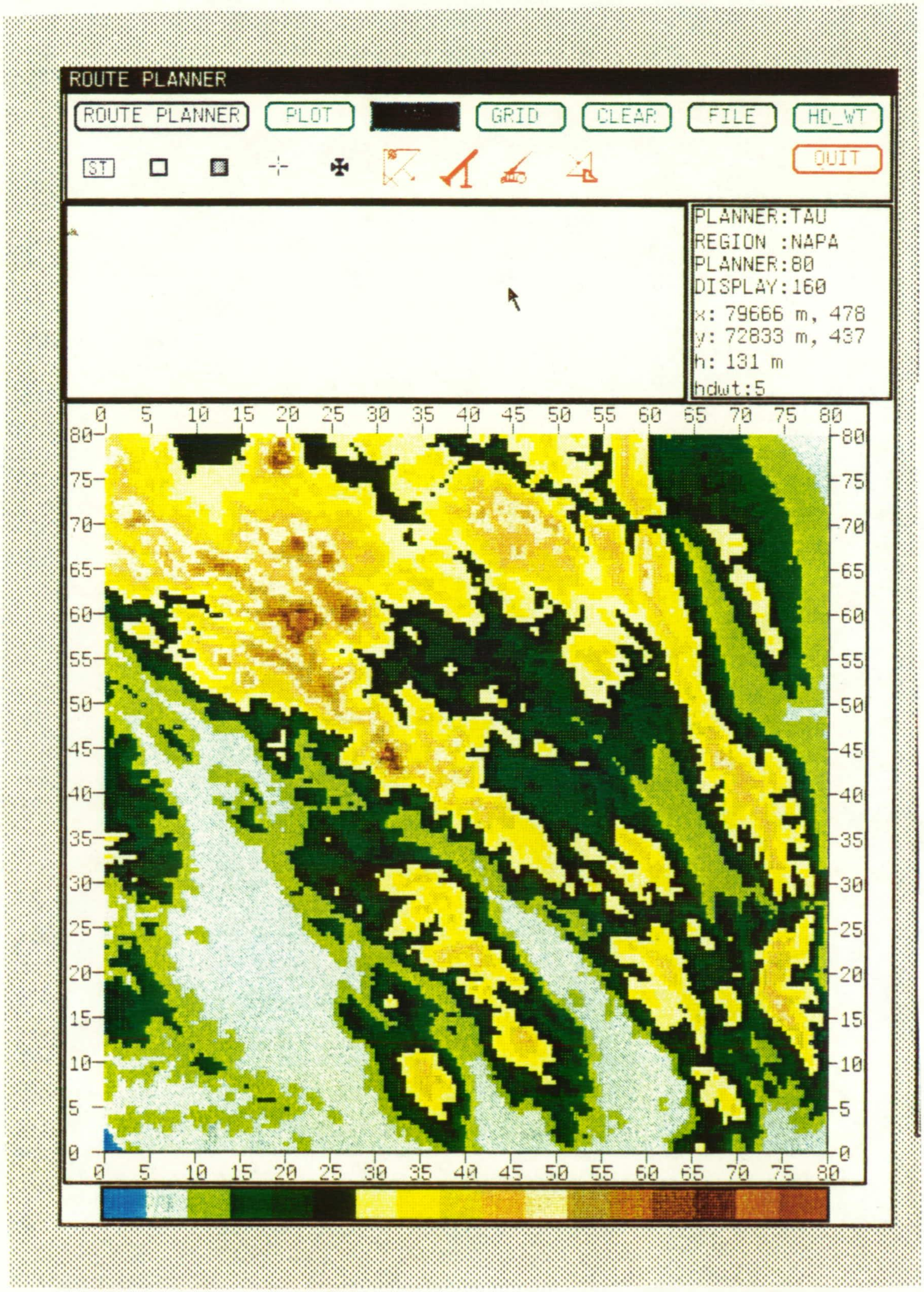

Figure 4. Graphical representation of 160 × 160 (0.5 km) grid terrain as used by route planner.

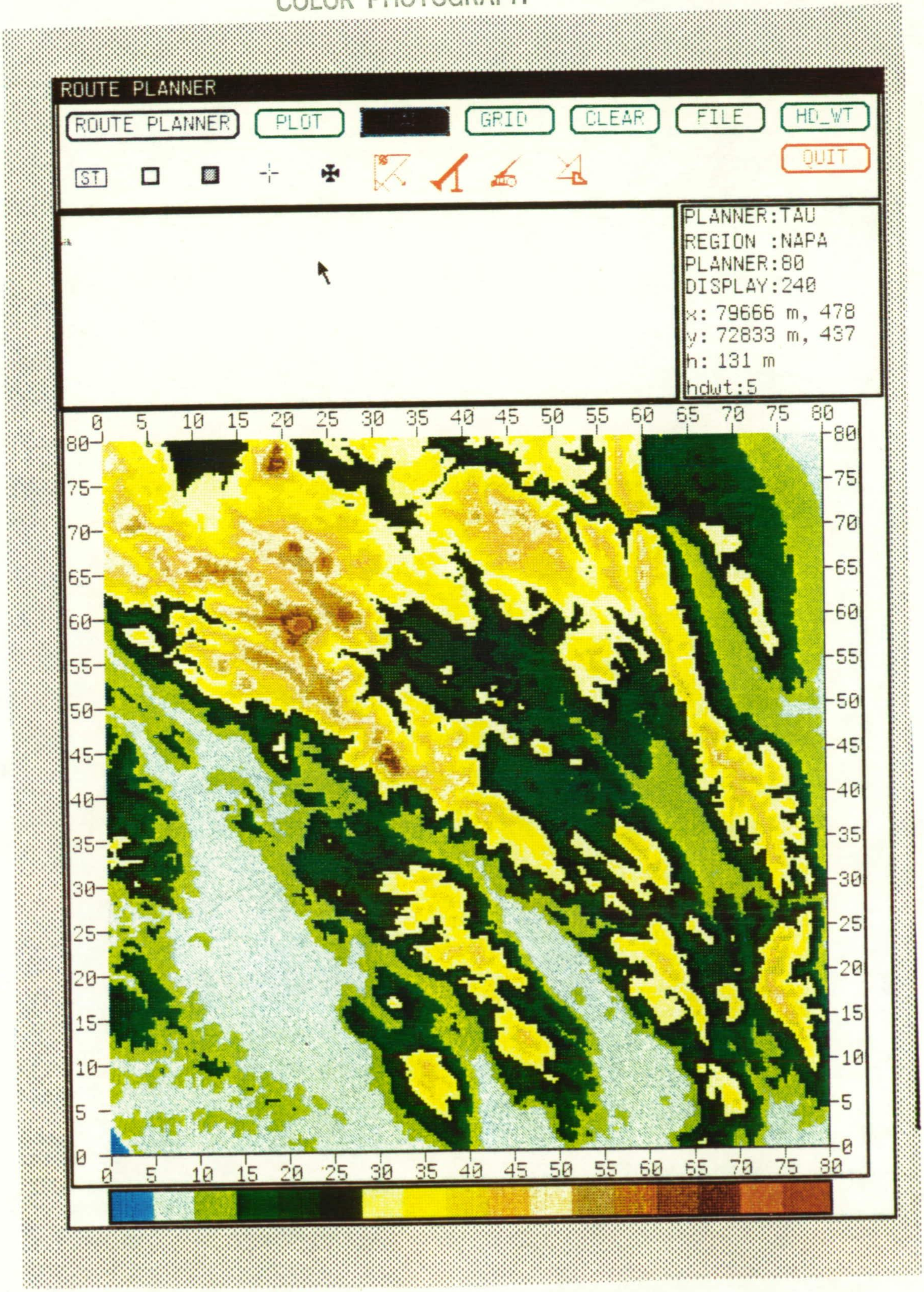

Figure 5. Graphical representation of 240 × 240 (0.33 km) grid terrain as used by route planner.

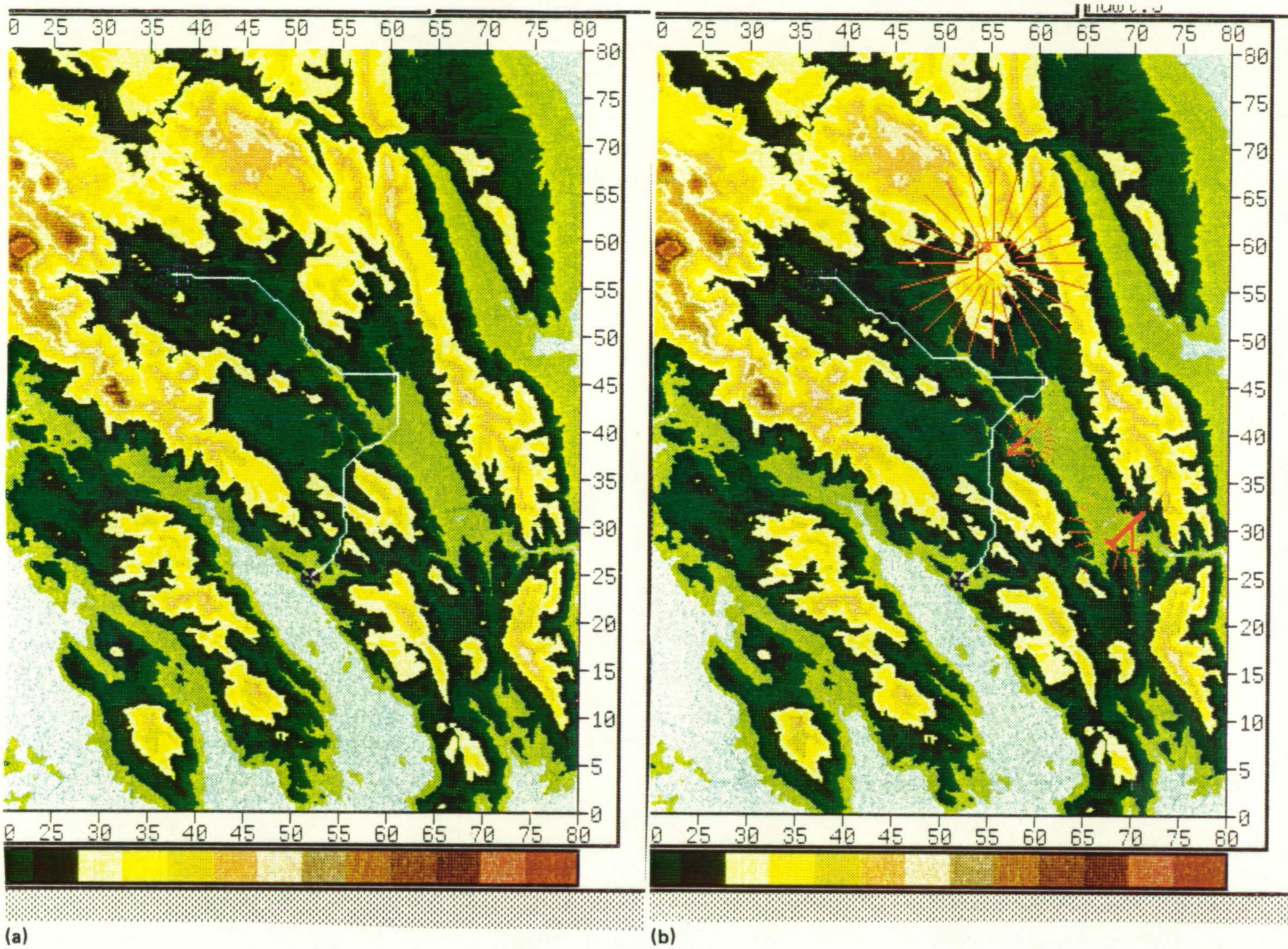

Figure 6. Threat examples and effects. (a) Route without threats, (b) route with threats.

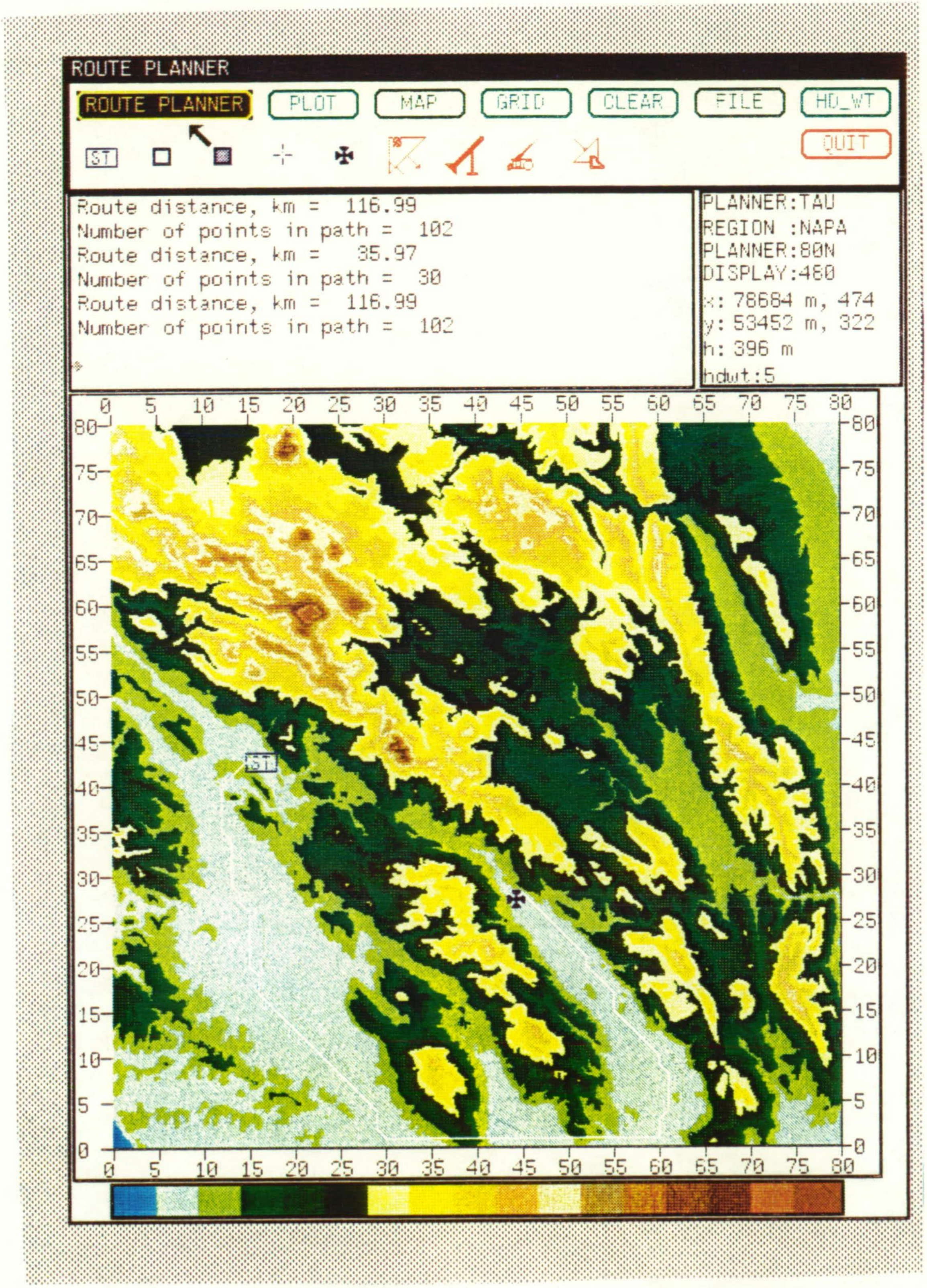

Figure 7. Example of valley following route using normalized terrain data.

Figure 8. Example of valley following route using non-normalized terrain data.

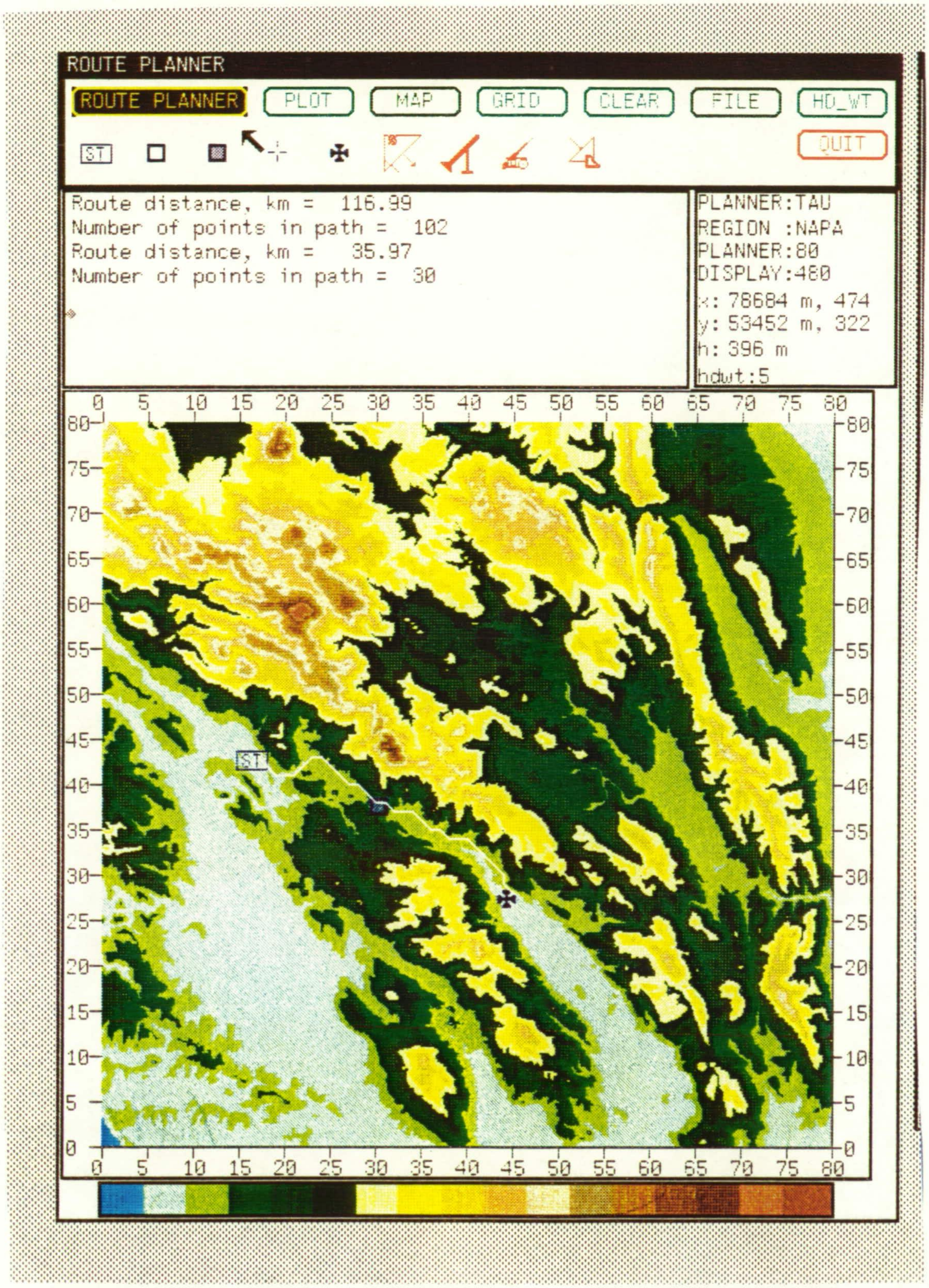

Figure 9. Route control using a control-point.

ORIGINAL PAGE
COLOR PHOTOGRAPH

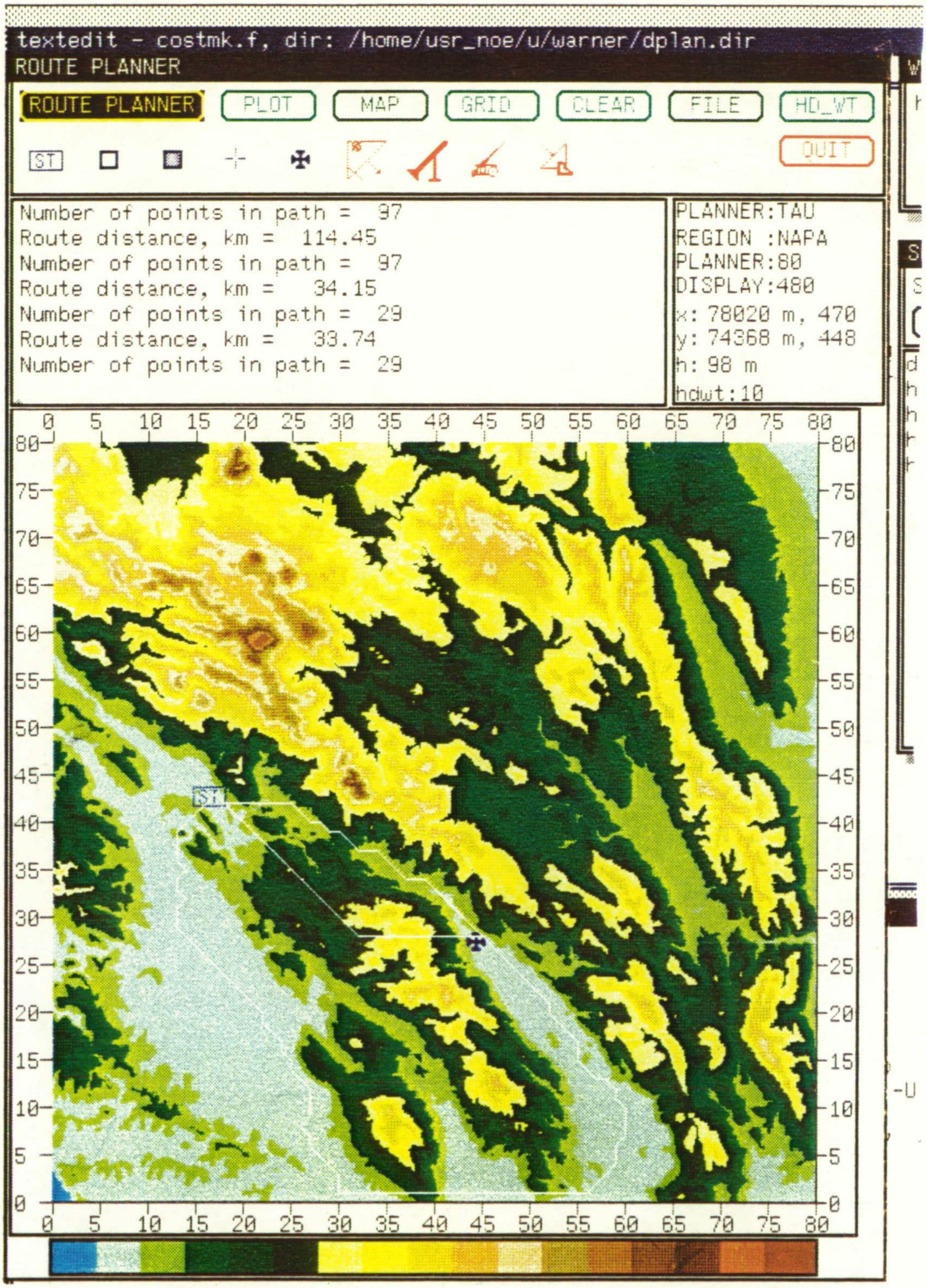

Figure 10. Route control using a cost function weighting factor.

PRECEDING PAGE BLANK NOT FILMED

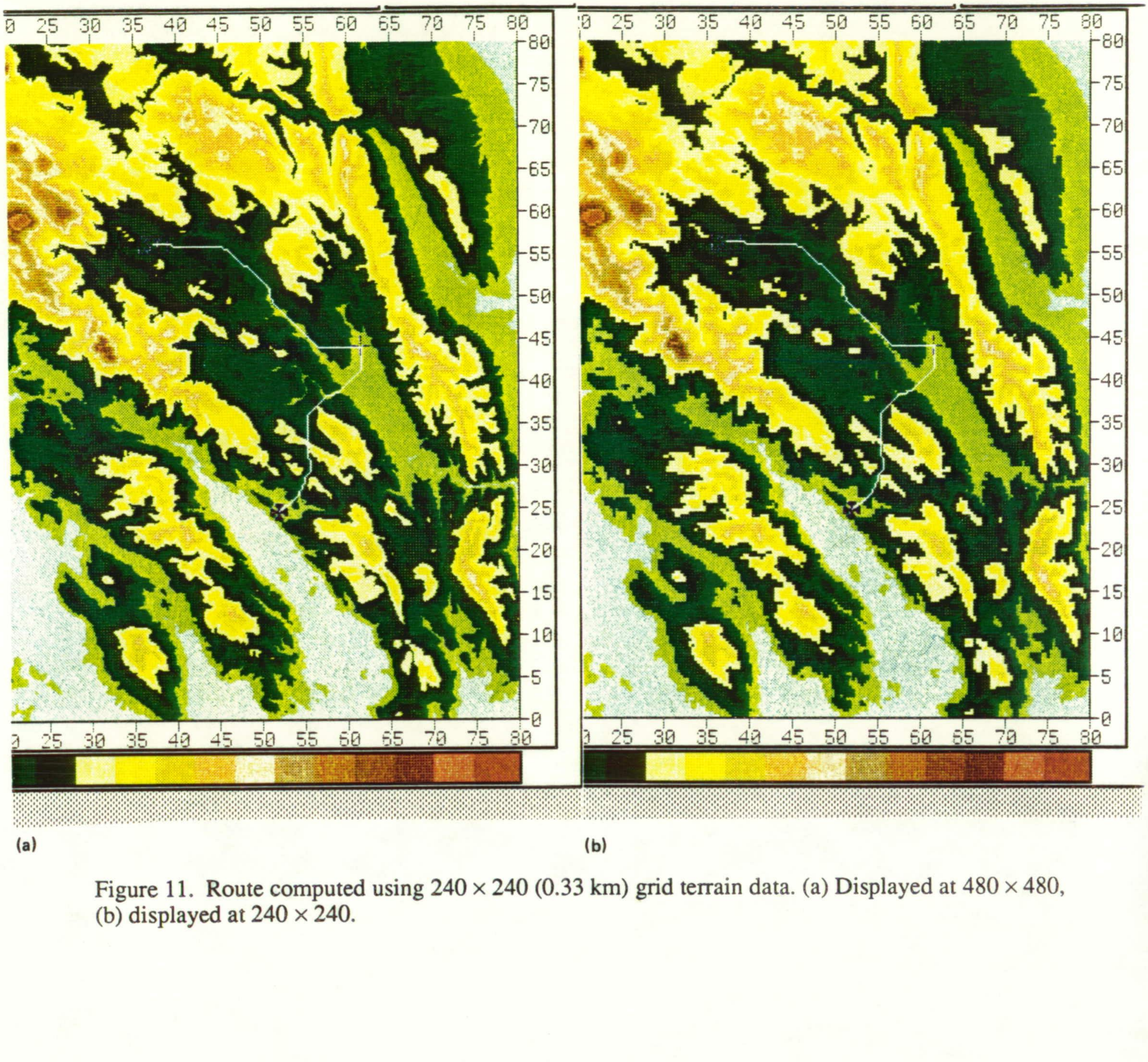

Figure 11. Route computed using 240 × 240 (0.33 km) grid terrain data. (a) Displayed at 480 × 480, (b) displayed at 240 × 240.

ORIGINAL PAGE
COLOR PHOTOGRAPH

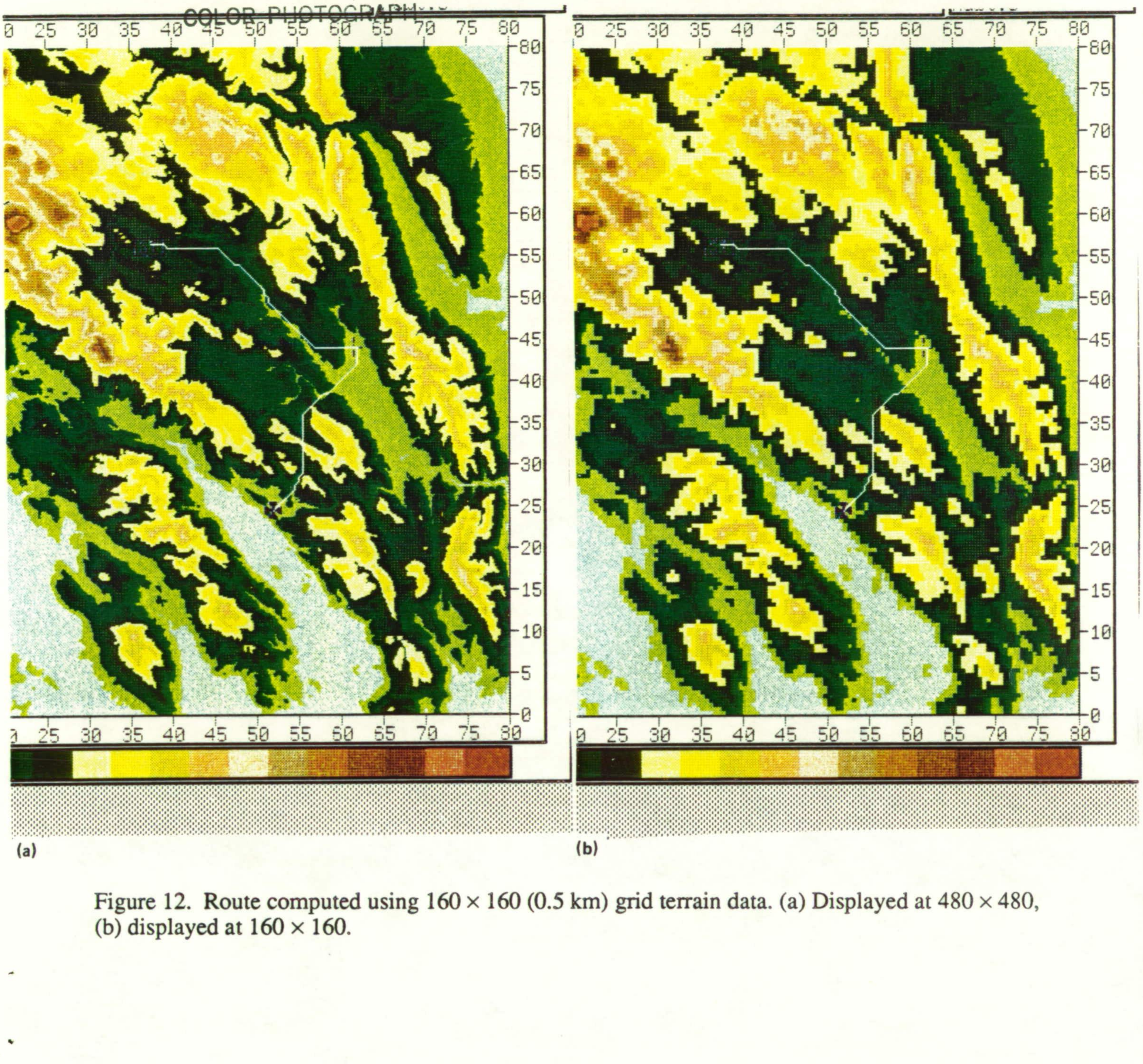

Figure 12. Route computed using 160 × 160 (0.5 km) grid terrain data. (a) Displayed at 480 × 480, (b) displayed at 160 × 160.

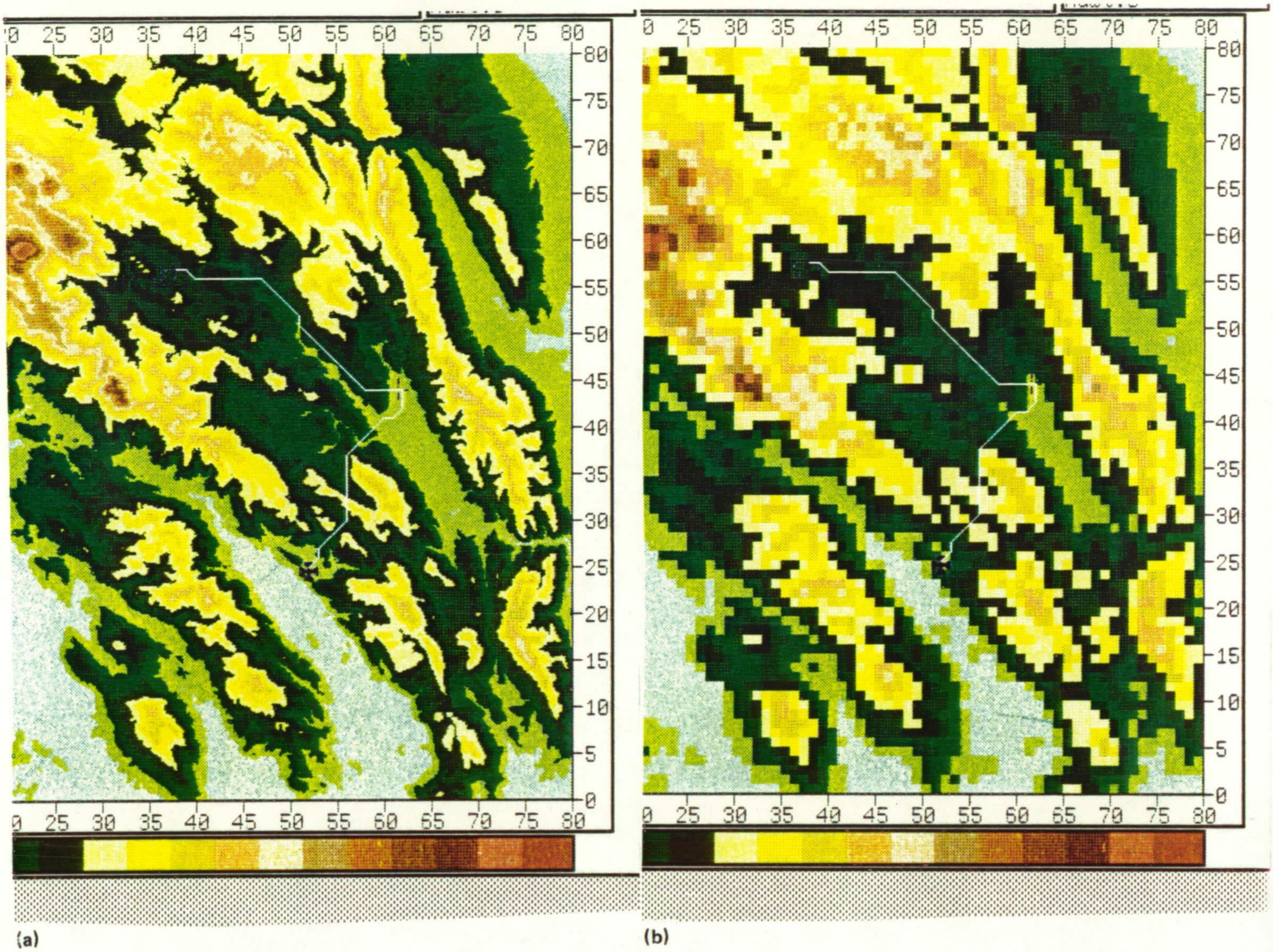

Figure 13. Route computed using 80 × 80 (1.0 km) grid terrain data. (a) Displayed at 480 × 480, (b) displayed at 80 × 80.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA TM-102882

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A Workstation-Based Evaluation of a Far-Field Route Planner for Helicopters

5. Report Date
June 1991

6. Performing Organization Code

7. Author(s)
David N. Warner, Jr. and Francis J. Moran

8. Performing Organization Report No.
A-91011

10. Work Unit No.
505-66-11

9. Performing Organization Name and Address
Ames Research Center
Moffett Field, CA 94035-1000

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes
Point of Contact: David N. Warner, Jr., Ames Research Center, MS 210-9,
Moffett Field, CA 94035-1000
(415) 604-5443 or FTS 464-5443

16. Abstract

Helicopter flight missions at very low, Nap of the Earth, altitudes place a heavy workload on the pilot. To aid in reducing this workload, Ames Research Center has been investigating various types of automated route planners. As part of an automated preflight mission planner, a route planner algorithm aids in selecting the overall (far-field) route to be flown. During the mission, the route planner can be used to replan a new route in case of unexpected threats or change in mission requirements. ~~This report describes~~ An evaluation of a candidate route planning algorithm, based on dynamic programming techniques, is described. This algorithm meets most of the requirements for route planning, both preflight and during the mission. In general, the requirements are to minimize the distance and/or fuel and the deviation from a flight time schedule, and must be flyable within the constraints of available fuel and time.

17. Key Words (Suggested by Author(s))
NOE flight
Mission management
Far-field planning
Waypoint path planning

18. Distribution Statement
Unclassified-Unlimited

Subject Category – 04

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
36

22. Price
A03